

Fahrenheit Co.

Communities

Eric Anderson - Proposed Product & How Product Solves Need
Christopher Bednarz - Customer Need & Existing Solutions
Zoe Chamlee - Technical Requirements
James W. Garrett - Budget, Revenue, Primary Editor
Mason Holley - Technical Challenges
Rishabh Tewari - Editor, Timeline

Date of Submission: September 13th, 2021

Project Proposal: Communities

Executive Summary

Communities is an app that allows people within the local scene to have an easier time to interact with each other. Our program does this by isolating the scope of the communities to the users' local areas. This app will consist of a map of the user's local area that will allow them to browse and join local groups and communities. There will also be a feed that will post all info related to the communities that the user joined.

Customer Need

Despite the internet's ability to connect people from all over the world to connect with one another, allowing anyone to find a community of people who share similar interests, many still find themselves feeling isolated and lonely. We believe these online communities are great for creating conglomerates of people who share similar interests all around the globe, but the problem is these communities are so large that individuality is ultimately lost. This creates an environment where personal and meaningful connections between individuals are unpropitious, in favor of posting whatever hot topic will allow someone to get the most upvotes. And when the only posts created are to win a popularity contest between thousands, the only connections made are shallow. These communities unwittingly sacrifice quality for quantity.

Existing Solutions

We've identified two types of competing products. The first are global services such as Discord and Reddit. Discord acts as a server based program where users can interact directly with each other using both text and a voice chat system. These servers are community made but there are servers for particular games or events. Reddit is a website that limits what you are able to post and where by topic via subreddits. Each subreddit has a theme of its own that is the "topic" of that particular area and is moderated by volunteer workers who check content to make sure that it follows community guidelines. Both of these programs offer ease of access to a wide and varying set of interests by breaking their products into many smaller communities.

However, while these solutions offer excellent connectivity between said communities, these communities both have a tendency to grow quite large and often consist of people that are located in increasingly distant parts of the world. This makes it harder for people to develop personal connections between others. Our product is designed to serve a more local scene.

The second type of similar apps are local based such as Meetup, Bumble, and Nextdoor. However, each one tends to have a more specific purpose or suffers from non distinct groupings. For instance, Nextdoor is very neighborhood-exclusive, Bumble is known better for dating, and Meetup has no restrictions on who can or can't join or post

Proposed Product

Our proposed product is an app that allows the user to view communities that are geographically close to his current location. Through our app, users will have the ability to browse the map of their city and select any communities that appeal to them. Once the communities are selected, the user will be able to see all of the posts, news, and events that are related to the communities.

How Product Solves Need

Our product solves the need by tightening the scope of the communities to the local scene while keeping the focus on freedom to select groups. In order to do this, we will be limiting the user's ability to post in subscribed, but distant, communities. Once a user has left the scope of a given community for an extended period of time, they will be put into a "Read Only" state that prevents them from posting. By limiting the scope, we believe that tighter knit communities will foster as locals will have an easier time connecting with other locals. By taking advantage of tailored groups, locals of an area should also find it easier to manage specific plans and events. Our design choices should help users stay up to date on what they specifically want and reduce the amount of fluff that they would see.

Technical Requirements

For this project our technical needs are as follows. Firstly is frontend, although our model for development should easily allow for a mobile app a web-app should at least exist, for this we will need a frontend framework which will most likely be React. Next is utilizing location data: our concept is heavily reliant upon the use of the user's location and will also call for mitigation of GPS spoofing to the greatest degree possible. We will also need a database as our platform will need to store data from both user posts and user accounts. This will probably be done with SQL, preferably postgres. Server hardware will be needed for hosting, we intend to use a Linux server which will need to be properly configured. For that hardware, we'll need webserver software to deploy the webapp, most likely nginx. Lastly is basic security requirements, HTTPS must be properly configured and applied for our web-app to meet modern security standards. HTTPS alone isn't enough however, we will also need common security optimizations like countermeasures and best development practices to prevent common web-app vulnerabilities like CSRF etc.

Technical Challenges

This project will present significant technical challenges that we must overcome in order to bring it to market. Gathering and making use of location based data, locking communities behind location based data, developing and deploying moderation tools, implementing automated spam protection, and automated account recovery each present their own unique challenges.

Gathering location based data will be solved through making use of the device's built in location system. If on a phone, we simply ask for access to the user's location through its systems. If on the webpage, we can ask through the browser's systems.

Using the location based data will be simple. Once we gather the location, we will save the location data as a coordinate. Cities will have a center coordinate that acts as their "center point". Anything within a certain radius of this point will act as communities that one can join for that city. Each user will also have a radius from their location data which will allow them to join the communities present in any cities they intersect.

Moderation tools must be implemented so the creators of communities have some level of control in their communities. If someone has broken the rules of a community, they can be removed from said community at a moderator's discretion. These moderation tools will simply

block that user from joining the community once removed. If needed, further methods such as IP banning can be implemented, along with a blacklist system.

Spam protection can be implemented by making use of a captcha. Features to detect abnormal levels of posts created in rapid succession can also be implemented to protect from spam. If an account goes over a certain number of posts in a given timeframe, the program will block them and they will be unable to write in any communities for a short time. Once that time has expired, if they continue to spam, they will be banned from posting. If our system fails and accidentally bans an innocent user, we can handle that through a ban appeal option.

Finally, account encryption and recovery is a necessary standard among all service based applications. We will store account details securely by encrypting and salting the data which will then be stored on our server. If a user requires their account information, they can send a request for that information through the app and will receive an identification email or text message. Once received and their identity validated, they will be able to receive their information.

Estimated Project Budget

In order to reach the goals set to create this program, our group estimated that It would take approximately 10 hours of coding and an additional 10 hours to reimplement features and iron out bugs in the programming. While this will obviously vary from member to member, we expect there to be a total of approximately 120 hours of work put into this program. Given that all group members are slated to graduate this upcoming semester, we will be using CS graduate salaries. According to the TTU Department of Computer Science, the average starting salary for BSCS graduates is around \$84,000 a year. At 2080 hours of work, this comes out to about a total of \$40.38/hour leading the project total of approximately \$4845.60. Aside from the time costs, we do not currently anticipate any additional expenses.

Revenue Model

In order to generate revenue with this project, we will be looking to implement a two-fold monetization system. First, we will put into a place a system for local businesses to advertise their businesses. We feel that this is a particularly good fit for our platform as it should help to encourage local businesses and strengthen local connections. It should also keep any ads for the users relatively relevant.

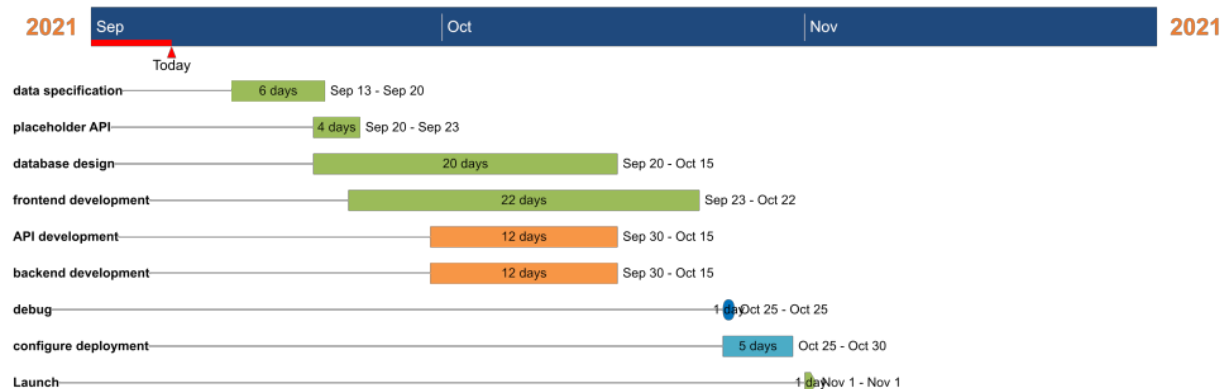
Secondly, we intend to implement a monthly subscription system where users can pay for "Premium" status. This Premium Status will allow the user to remove ads from their app and post in distant communities that they would otherwise be unable to speak in.

Given our current estimate for product cost, we expect the app to take 20-24 months to turn a profit. Our current estimates have us starting at around \$100 of revenue a month and increasing to around triple that by the 20 month mark averaging to about \$200 per month.

Estimated Timeline

The estimated timeline of our project includes the work to be done in three phases: Planning, Coding, and Improvements & Debugging. The first phase consists of brainstorming the project, designing specifications, and deciding on what the minimum viable product consists

of along with finalizing on the details of map placement, the feed, and search function, etc. In this stage, we intend to finalize how we intend to implement our baseline features. Following this, we enter our second phase, which consists of working on the development of our application, where we develop our MVP, and make it functional according to the requirements set in Phase 1. We will initially be focusing on getting our baseline project out but, once we have hit our requirements, we may take a look into improving the app further. Once all the required features have been implemented, we will begin the third stage where we will deal with error checking, and bugfixes along with any other problems that the app may encounter.



List of Deliverables

Deliverables for the app are as follows:

- Database that tracks all of the users, communities, and posts.
- Account creation and login system with username and password
- Website and App front-end
- Feed consisting of the news, posts, and events of the user's associated communities
- Map for the user to scroll around

References

Department of Computer Science. (2019). *Department of Computer Science*. Texas Tech University Departments. Retrieved September 12, 2021, from https://www.depts.ttu.edu/cs/includes/CS_department_sheet.pdf.

CARLILE, LIZ. *Development Online: Making the Most of Social Media*. International Institute for Environment and Development, 2011, www.jstor.org/stable/resrep01460. Accessed 13 Sept. 2021.